Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

1. (Original) A system for handling transport protocol segments (TPSes), comprising:

a receiver that receives an incoming TPS, the incoming TPS comprising an aligned upper layer protocol (ULP) header and a complete ULP data unit (ULPDU),

wherein the receiver directly places the complete ULPDU into a host memory.

2. (Original) The system according to claim 1,

wherein the receiver comprises a network subsystem and the host memory,

wherein the network subsystem receives the incoming TPS and directly places data of the complete ULPDU into the host memory.

3. (Original) The system according to claim 1, wherein the network subsystem comprises a network interface card (NIC) or a network controller.

4. (Original) The system according to claim 1, wherein the ULPDU comprises a framing protocol data unit (FPDU).

5. (Original) The system according to claim 4, wherein the FPDU comprises a data unit created by a ULP using a marker-based ULPDU aligned (MPA) framing protocol.

6. (Original) The system according to claim 1, wherein the aligned ULP header comprises an aligned FPDU header.

7. (Original) The system according to claim 6, wherein the aligned ULP header

comprises the aligned FPDU header disposed adjacently to a TPS header of the TPS.

8. (Original) The system according to claim 1, wherein the aligned ULP header is disposed a preset length away from a TPS header of the TPS.

9. (Original) The system according to claim 1, wherein the aligned ULP header is disposed a particular length away from the TPS header, the particular length being related to information in a field in the TPS.

10. (Original) The system according to claim 9, wherein the field comprises a marker field.

11. (Original) The system according to claim 1, wherein the receiver is a flow-through receiver.

12. (Original) The system according to claim 1, wherein the TPS comprises a transmission control protocol (TCP) segment.

13. (Original) The system according to claim 12, wherein the TCP segment is part of a TCP byte stream.

14. (Original) The system according to claim 1,

wherein the receiver comprises a buffer, and

wherein the size of the buffer does not scale approximately linearly with a network speed or a network bandwidth.

15. (Original) The system according to claim 1,
wherein the receiver comprises a buffer, and
wherein the size of the buffer does not scale with the number of connections.

16. (Original) The system according to claim 1, wherein the incoming TPS comprises information that is used to place the complete ULPDU in the host memory.

17. (Original) The system according to claim 1, wherein the receiver does not store partial cyclical redundancy check (CRC) values.

18. (Original) The system according to claim 1, wherein the incoming TPS comprises an out-of-order incoming TPS.

19. (Original) The system according to claim 1, wherein the receiver does not store only a portion of the complete ULPDU.

20. A system for handling TPSes, comprising:
a sender that sends a TPS, the sent TPS comprising an aligned ULP header and one or more complete ULPDUs.

21. (Original) A method for handling TPSes, comprising:
aligning an FPDU header in a known position in a TPS with respect to a TPS header; and
placing a complete FPDU in the TPS.

22. (Original) A method for handling TPSes, comprising:
receiving an incoming TPS, the TPS comprising a complete FPDU and an FPDU header

in a known position with respect to a TPS header.

23. (Original) The method according to claim 22, wherein the FPDU header is adjacent to the TPS header.

24. (Original) The method according to claim 22, further comprising:

performing layer 2 (L2) processing, layer 3 (L3) processing and layer 4 (L4) processing on the incoming TPS via a network subsystem.

25. (Original) The method according to claim 24, further comprising:

obtaining FPDU length information from the FPDU header.

26. (Original) The method according to claim 25, further comprising:

programming a direct memory access (DMA) engine to copy data of the FPDU from the network subsystem to a host memory.

27. (Original) The method according to claim 26, further comprising:

programming the DMA engine to move FPDU through a cyclical redundancy checking (CRC) machine.

28. (Original) The method according to claim 22, wherein the TPS comprises a plurality of complete FPDUs.

29. (Original) The method according to claim 24, further comprising:

performing ULP processing on the incoming TPS via the network subsystem,

wherein the L2 processing, the L3 processing, the L4 processing and the ULP processing

can occur in parallel or in any order.

30. (Original) The method according to claim 29,

wherein the L2 processing, the L3 processing, the L4 processing and the ULP processing do not occur in the listed order in a receiver.

31. (Original) The method according to claim 29,

wherein the ULP processing, the L4 processing, the L3 processing and the L2 processing do no occur in the listed order in a transmitter.

32. (Original) A system for handling transport protocol segments (TPSes), comprising:

a receiver comprising a direct memory access (DMA) engine,

wherein the receiver receives an incoming TPS, the incoming TPS comprising an aligned upper layer protocol (ULP) header and a complete ULP data unit (ULPDU),

wherein the receiver programs the DMA engine once to place the complete ULPDU into a host memory.

33. (Original) The system according to claim 32,

wherein the receiver comprises a cyclical redundancy check (CRC) machine, and

wherein the receiver uses the CRC machine once per ULPDU.

34. (Original) The system according to claim 33,

wherein the receiver comprises a non-flow-through network interface card (NIC), and

wherein the DMA engine and the CRC machine are part of the non-flow-through NIC.

35. (Original) The system according to claim 34, wherein the non-flow-through NIC

comprises a local memory.

36. (Original) The system according to claim 35, wherein the non-flow-through NIC performs a CRC calculation before or as the complete ULPDU is stored in the local memory.

37. (Original) The system according to claim 35, wherein the non-flow-through NIC performs a CRC calculation after the complete ULPDU is stored in the local memory.

38. (Original) The system according to claim 35, wherein the non-flow-through NIC performs a CRC calculation during a process by which the complete ULPDU is sent from the local memory to a host memory.

39. (Original) The system according to claim 35, wherein the complete ULPDU comprises a marker-aligned protocol data unit.

40. (Original) The system according to claim 33,
wherein the receiver comprises a flow-through NIC, and
wherein the DMA engine and the CRC machine are part of the flow-through NIC.

41. (Original) The system according to claim 40, wherein the flow-through NIC comprises a buffer.

42. (Original) The system according to claim 41, wherein the non-flow-through NIC performs a CRC calculation before or as the complete ULPDU is stored in the buffer.

43. (Original) The system according to claim 41, wherein the CRC calculation is a ULP

CRC calculation.

44. (Original) The system according to claim 40, wherein the complete ULPDU comprises a marker-aligned protocol data unit.

45. (Original) A method for handling TPSes, comprising:

(a) receiving an incoming TPS, the TPS comprising a complete FPDU and an FPDU header in a known position with respect to a TPS header

(b) performing layer 2 (L2) processing on the incoming TPS;

(c) performing layer 3 (L3) processing on the incoming TPS;

(d) performing layer 4 (L4) processing on the incoming TPS; and

(e) performing ULP processing on the incoming TPS,

wherein the performing of (b), (c), (d) and (e) occurs in any order.

46. (Original) The method according to claim 45, wherein at least two of the performing of (b), (c), (d) and (e) occurs concurrently.